OIPE
MAR 31 2005
PATENT & TRADEMARK OFFICE JC11

AF IDW
2127

IBM Docket No. **POU919990100US1**

Certificate of Mailing/Facsimile 37 CFR §1.8(a)

I hereby certify that this correspondence is being:

__X__ deposited with the United States Postal Service as first class mail in an envelope with sufficient postage addressed to the:

____ facsimile transmitted to **[Fax Number]** to the:

Mail Stop Appeal Brief - Patents
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on March 28, 2005.

Susan L. Phelps

Signature

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant: Tuel et al.	:	GROUP ART UNIT: 2127 Conf. No. 9151
Serial No.: 09/588,492	:	Examiner: M. Banankhah
Filed: June 6, 2000	:	March 28, 2005
Title: THREAD DISPATCHER FOR	:	Lawrence D. Cutter
MULTI-THREADED COMMUNICATION	:	IBM Corporation
LIBRARY	:	2455 South Road, M/S P386
	:	Poughkeepsie, NY 12601

APPEAL BRIEF UNDER 37 C. F. R. § 1.192

BRIEF OF APPELLANTS

Hon. Commissioner for Patents
Mail Stop Appeal Brief-Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is an appeal from a final rejection dated October 27, 2004 rejecting Claims 1 through 48 which are all of the claims being considered in the above-identified patent application. This brief is accompanied by a transmittal letter authorizing the charging of appellants' deposit account for payment as set forth in 37 C.F.R. § 1.17(c).

04/01/2005 HALI11 00000001 090463 09588492
01 FC:1402 500.00 DA

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by virtual of an assignment executed on June 6, 2000, by the inventors herein and recorded in the United States Patent and Trademark Office at Reel No. 010894 and Frame No. 0806 on June 6, 2000. Therefore, the real party in interest is International Business Machines Corporation.

II. RELATED APPEALS AND INTERFERENCES

To the knowledge of the appellants, appellants' undersigned legal representative and the assignee, there are no other appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the instant appeal.

III. STATUS OF CLAIMS

This patent application was filed on June 6, 2000, with the United States Patent and Trademark Office. The status of the claims in the present application is as follows:

Claims Allowed: None;
Claims Objected to: None;
Claims Rejected: 1 through 48;
Claims Canceled: None.

Accordingly, appellants hereby appeal the rejection of Claims 1 through 48.

IV. STATUS OF AMENDMENTS

Applicants received a final rejection on October 27, 2004. In response thereto, applicants filed an amendment on January 26, 2005, within the three-month statutory period permitted. Since the response under 37 CFR § 1.116 was not submitted within the two-month time period, applicants do not anticipate receiving an advisory action from the examiner. In particular, it is

noted that no advisory action has been received by the applicants from the examiner. A call to the examiner to ascertain the status of the amendment was not returned. Accordingly, at this point in time, it is unknown whether or not the examiner will enter applicants' amendment under Rule 116 for purposes of this appeal. Accordingly, it is applicants' assumption that this amendment will not be entered. However, it is noted that the amendment submitted under Rule 116 was in response to a rejection under 35 U.S.C § 112. It is also noted that the Rule 116 response changed only one word in each of independent Claims 1, 13, 25 and 37. It is further noted that this amendment was made for clarifying purposes only.

Accordingly, in light of the above, it is assumed that the status of the amendment is such that the claims are in the same state as they were prior to the submission of the response under Rule 116 submitted on January 26, 2005.

V. SUMMARY OF THE INVENTION

Most particularly for purposes of this Appeal, there is provided a method for efficiently dispatching threads which are awaiting messages in a multi-threaded communication library. In particular, in applicants' invention, threads are preassigned to messages which are to be received. In applicants' invention, those threads whose assigned messages have not yet been received are put to sleep, that is, placed in an inactive status. Upon receipt of a message awaking its preassigned thread, the wakened thread begins executing its message processing function. Furthermore, and most importantly for the present invention, this awakened thread not only processes the received message, it also tests to see if any other threads are ready to run. In particular, it is seen that the novel aspect of the present invention is that a thread which normally just processes its own assigned received message is now provided with an additional function namely the testing to see whether or not any other threads are ready to run. This is efficient in that this task of testing to see if any other threads are ready is something which is ordinarily handled by an entirely separate thread. In order to carry out the operation of this second thread, various operations relating to starting and initializing the thread have to be undertaken by the data processing system. However, in the present invention, this function is carried out by the awakened thread associated with the preassigned message.

VI. ISSUES

(A) Whether Claims 1 through 48 and in particular, independent Claims 1, 13, 25 and 37 are indefinite under 35 U.S.C. § 112.

(B) Whether Claims 1 through 48 are obvious under 35 U.S.C. § 103 based upon the patent to Huff et al. (US Patent No. 6,457,064) in view of the document titled "UNIX Internals The New Frontiers" by Uresh Vahalia.

VII. GROUPING OF CLAIMS

No claim grouping is asserted at this time.

VIII. ARGUMENTS

A. Issue #1 Rejection under 35 U.S.C. § 112

In this rejection, the examiner has pointed to applicants' independent Claims 1, 13, 25 and 37 in that they recite "executing said awakened thread which processes the received message and tests to see if any threads are ready to run." In this respect, it is noted that the quotation from applicants' claims as stated in the examiner's Final Rejection is in error in that it states "test to see if any thread are" where the actual text recites "tests to see if any threads are ready to run." Nonetheless, the examiner goes on to assert that this recitation is vague because it is unclear why the thread that is awakened and executing is being tested to see whether it is ready to run. Clearly, this is not the interpretation that is to be given to applicants' claims.

In this regard, it is noted that applicants have, in their response filed under Rule 116, offered to insert the word "other" so that applicants' recitation, for example, in the last step of Claim 1 recites: "executing said awakened thread which processes the received message and

tests to see if any other threads are ready to run." Clearly, it makes no sense for a thread which is executing and which has been awakened to test itself to see if it is ready to run since it is clearly already running. It is therefore the natural and only logical conclusion to be drawn is that the testing done by the awakened thread is the testing of other threads to see if they are ready to run. It is this testing by the awakened thread which is unique and not obvious in applicants' claim. No separate polling thread needs to be launched. Since the launching of a new thread is a time-consuming operation, its avoidance in the present invention is thereby seen to provide computational advantages in the message processing arts.

The very fact that the examiner is questioning why the threads that are awakened and executing are operating to perform test operations to determine whether or not readiness to run is possible undermines the examiner's very arguments. In particular, the examiner's statement lends credence to applicants' position that the claims, as currently written, are not in any way vague or indefinite. In particular, to those of ordinary skill in the art reading applicants' last recited claim step, it would be abundantly clear that the testing for readiness is performed on other threads. It is logically impossible and therefore unreasonable to read applicants' recited claim steps as an indication that the awakened thread is testing itself to see if it is ready to run which is an empty gesture since it is clearly already running. It is therefore seen that the only logical interpretation of applicants' claim language is that the awakened thread is employed to test whether or not other threads are ready to run.

Accordingly, while it is noted that applicants have indicated their willingness to inject the word "other" into the recited claims, the examiner has not yet expressed his willingness to accept this change. Accordingly, applicants assert that even as they currently stand, the claims are neither vague nor indefinite and that they do in fact teach those of ordinary skill in the art that the awakened thread tests other threads to see if they are ready to be run.

Accordingly, it is seen that Claims 1 through 48 and in particular, independent Claims 1, 13, 25 and 37 are not in fact neither vague nor indefinite. Properly read, the action carried out in the final claim step is well understood and clear to those with ordinary skill in the art.

Accordingly, it is therefore respectfully requested that the rejection under 35 U.S.C. § 112 be reversed.

B. Issue # 2 Rejection under 35 U.S.C. § 103

Attention is now directed to the rejection applicants' Claims 1 through 48 under 35 U.S.C. § 103 based upon the patent to Huff et al., in further view of the article by Vahalia.

On Page 3 of the final rejection, the examiner lays out the teachings of Huff et al. Nowhere in the argument which the examiner employs to reject applicants' claims is there any mention of a step in which an awakened and executing thread processes a received message and further tests to see if there are any threads that are ready to be run. The examiner's arguments with respect to applicants' independent claims as set forth on page 3 of the final rejection are solely directed to applicants' claim step 2 which recites: "putting to sleep, those threads whose assigned messages have not been received." There is no reference or allusion to applicants' claim step 4.

On Page 4 of the aforementioned Final Rejection, the examiner quotes applicants' arguments as set forth in applicants' response dated June 30, 2004. Therein, applicants argued as follows:

> "Accordingly, particularly as amended, applicants' present claims provide a method for dispatching threads in which the thread that is awakened resumes polling after it has processed the message. Thus, in the claimed invention there is no longer a need for a single dedicated polling thread. In contrast, polling is now provided as an activity which is passed from thread to thread as threads are awakened. This saves a context switch back to the polling thread when a message has been processed."

The examiner argues that since the words in applicants' claims do not specifically recite that polling activity is "passed from one thread to another," that this does not in fact happen. However, it is not necessary that the natural consequences of an applicant's method have to be recited *in haec verba* in order to support an applicant's claims. This passing of polling activity

from one thread to another is a natural consequence of the action that takes place in applicants' recited claims, particularly as seen in step 4 of Claim 1, for example. That these consequences naturally flow from the recited language means that it is not essential, necessary or even desirable for this aspect to be specifically recited in applicants' claims for the simple reason that this result is a natural consequence of the operation of applicants' claimed method.

Moreover, it is noted that applicants' fourth claim step in Claim 1 stands out as being exemplary and particularly points out the differences between applicants' claimed invention and the art cited. As described above, in this last claim step, in which a thread is awakened, its execution not only processes the message which has been assigned to it but is a step in which the awakened thread goes on to do other things. There is no teaching, disclosure or suggestion in either of the two cited documents that would lead one of ordinary skill in the art that an awakened thread which is designed for message processing does anymore than just that, namely, process messages. There is no teaching, disclosure or suggestion that such a thread is usable to test the condition of other threads to determined whether they are ready to be processed.

In extremely stark contrast to the specific teaching and recitation of applicants' claims, the patent to Huff et al. specifically teaches and requires that a separate polling thread be initiated. In point of fact, it is applicants' intent to avoid the initialization of yet another thread together with its concomitant overhead and disadvantages. Why try to accomplish something with a new thread when an existing thread is already executing and which can be designed to carry out the intended functions. It is in this way that message processing is streamlined and made more efficient.

Those of ordinary skill in the art, following the teachings of Huff et al., would be required (not just inclined but required) to initiate a new polling thread. This is a step which is specifically taught against in applicants' specification and is avoided in applicants' claims. When cited art teaches against that which is claimed, cited art cannot in any way be employed as a basis for a rejection of an applicant's claims. Accordingly, it is seem that the rejection of applicants' Claims 1 through 48 based on the two cited documents cannot be supported. It is therefore

respectfully requested that the rejection of applicants' Claims 1 through 48 under 35 U.S.C. § 103 based on the two cited documents be reversed.

CONCLUSIONS

In light of the above, the present applicants respectfully request a reversal of the rejection of Claims 1 through 48 under 35 U.S.C. § 112. Likewise, applicants respectfully request a reversal of the rejection of applicants' Claims 1 through 48 under 35 U.S.C. § 103 based upon two cited documents of record. For all of the above reasons, applicants traverse the rejections of their claims. It is, moreover, respectfully requested that reversal of all rejections be granted.

APPENDIX

1. A method for efficiently dispatching threads awaiting messages in a multi-threaded communication library comprising:

pre assigning threads to messages to be received;

putting to sleep, those threads whose assigned messages have not been received;

upon receipt of a message, awakening its pre assigned thread; and

executing said awakened thread which processes the received message and tests to see if any threads are ready to run.

2. The method of claim 1 wherein the selection of the thread to be dispatched is based on its priority as set when the thread is put to sleep.

3. The method of claim 1 wherein said preassigning threads step comprises:

creating a thread-specific structure for each thread, each thread-specific structure having a ready flag and a condition variable unique to its preassigned thread;

creating a handle for each message to be received; and

having a thread invoke message passing logic for a particular handle, thereby associating the thread and the message.

4. The method of claim 3 wherein said putting to sleep step comprises:

enqueing for a received message, a preassigned thread-specific structure into a first queue;

writing into said handle associated with the message received, an identification of said thread-specific structure enqued for the received message, and

placing said thread-specific structure for the received message in the WAIT condition.

5. The method of claim 4 wherein said awakening step comprises;

completing said received message;

changing the condition of the thread-specific structure for the completed received

structure to the READY condition; and

dequeueing with a queue manager, the next thread-specific structure in said first queue in the READY condition and sending its thread a thread awakening condition signal.

6. The method of claim 5 further comprising;

allocating in said preassigning step, buffer space for storing messages to be received; and

in said putting to sleep step, identifying in said handle the buffer in which the message associated with the handle is to be stored when it is received.

7. The method of claim 6 wherein said completing said received message comprises storing said received message in the buffer identified in the associated handle for the received message.

8. The method of claim 5 wherein said queue manager dequeues the next thread-specific structure using a First-In-First-Out policy.

9. The method of claim 5 wherein said queue manager dequeues the next thread-specific structure using a Last-In-First-Out policy.

10. The method of claim 5 wherein said queue manager dequeues the next thread-specific structure based on a priority value contained in said structure.

11. The method of claim 5 further comprising obtaining a lock for the handle associated with said received message such that the awakened thread may process only the received message.

12. The method of claim 11 further comprising releasing said lock after said awakened thread has processed said received message such that said awakened thread may continue with other work.

13. A computer program product comprising a computer useable medium having computer readable program code means therein for efficiently dispatching threads awaiting messages in a multi-threaded communication library, said computer readable program code means in said computer program product comprising:

computer readable program code means for preassigning threads to messages to be received;

computer readable program code means for putting to sleep, those threads whose assigned messages have not been received;

computer readable program code means for, upon receipt of a message, awakening its preassigned thread; and

computer readable program code means for executing said awakened thread, processing the received message and testing to see if any threads are ready to run.

14. The computer program product of claim 13 wherein the selection of the thread to be dispatched is based on its priority as set when the thread is put to sleep.

15. The computer program product of claim 13 wherein said computer readable program code means for preassigning threads comprises:

computer readable program code means for creating a thread-specific structure for each thread, each thread-specific structure having a ready flag and a condition variable unique to its preassigned thread;

computer readable program code means for creating a handle for each message to be received; and

computer readable program code means for having a thread invoke message passing logic for a particular handle, thereby associating the thread and the message.

16. The computer program product of claim 15 wherein said computer readable program code means for putting to sleep comprises:

computer readable program code means for enqueing for a received message, a preassigned thread-specific structure into a first queue;

computer readable program code means for writing into said handle associated with the message received, an identification of said thread-specific structure enqued for the received message, and

computer readable program code means for placing said thread-specific structure for the received message in the WAIT condition.

17. The computer program product of claim 16 wherein said computer readable program code means for awakening comprises;

computer readable program code means for completing said received message;

computer readable program code means for changing the condition of the thread-specific structure for the completed received structure to the READY condition; and

computer readable program code means for dequeueing with a queue manager, the next thread-specific structure in said first queue in the READY condition and sending its thread a thread awakening condition signal.

18. The computer program product of claim 17 further comprising;

computer readable program code means for allocating in said preassigning step, buffer space for storing messages to be received; and

said computer readable program code means for putting to sleep includes, computer readable program code means for identifying in said handle the buffer in which the message associated with the handle is to be stored when it is received.

19. The computer program product of claim 18 wherein said computer readable program code means for completing said received message comprises computer readable program code means for storing said received message in the buffer identified in the associated handle for the received message.

20. The computer program product of claim 17 wherein said queue manager includes computer readable program code means for dequeueing the next thread-specific structure using a First-In-First-Out policy.

21. The computer program product of claim 17 wherein said queue manager includes computer readable program code means for dequeueing the next thread-specific structure using a Last-In-First-Out policy.

22. The computer program product of claim 17 wherein said queue manager includes computer readable program code means for dequeueing the next thread-specific structure based on a priority value contained in said structure.

23. The computer program product of claim 17 further comprising computer readable program code means for obtaining a lock for the handle associated with said received message such that the awakened thread may process only the received message.

24. The computer program product of claim 23 further comprising computer readable program code means for releasing said lock after said awakened thread has processed said received message such that said awakened thread may continue with other work.

25. An apparatus for efficiently dispatching threads awaiting messages in a multi-threaded communication library comprising:

means for preassigning threads to messages to be received;

means for putting to sleep, those threads whose assigned messages have not been received;

means for, upon receipt of a message, awakening its preassigned thread; and

executing said awakened thread which processes the received message and tests to see if any threads are ready to run.

26. The apparatus of claim 25 wherein the selection of the thread to be dispatched is based on its priority as set when the thread is put to sleep.

27. The apparatus of claim 25 wherein said means for preassigning threads comprises:

means for creating a thread-specific structure for each thread, each thread-specific structure having a ready flag and a condition variable unique to its preassigned thread;

means for creating a handle for each message to be received; and

means for having a thread invoke message passing logic for a particular handle, thereby associating the thread and the message.

28. The apparatus of claim 27 wherein said means for putting to sleep comprises:

means for enqueing for a received message, a preassigned thread-specific structure into a first queue;

means for writing into said handle associated with the message received, an identification of said thread-specific structure enqued for the received message, and

means for placing said thread-specific structure for the received message in the WAIT condition.

29. The apparatus of claim 28 wherein said means for awakening comprises;

means for completing said received message;

means for changing the condition of the thread-specific structure for the completed received structure to the READY condition; and

means for dequeueing with a queue manager, the next thread-specific structure in said first queue in the READY condition and sending its thread a thread awakening condition signal.

30. The apparatus of claim 29 further comprising;

means for allocating in said preassigning step, buffer space for storing messages to be received; and

in said means for putting to sleep, means for identifying in said handle the buffer in which the message associated with the handle is to be stored when it is received.

31. The apparatus of claim 30 wherein said means for completing said received message comprises means for storing said received message in the buffer identified in the associated handle for the received message.

32. The apparatus of claim 29 wherein said queue manager includes means for dequeueing the next thread-specific structure using a First-In-First-Out policy.

33. The apparatus of claim 29 wherein said queue manager includes means for dequeueing the next thread-specific structure using a Last-In-First-Out policy.

34. The apparatus of claim 29 wherein said queue manager includes means for dequeueing the next thread-specific structure based on a priority value contained in said structure.

35. The apparatus of claim 29 further comprising means for obtaining a lock for the handle associated with said received message such that the awakened thread may process only the received message.

36. The apparatus of claim 35 further comprising means for releasing said lock after said awakened thread has processed said received message such that said awakened thread may continue with other work.

37. An apparatus comprising:

a data processing system;

a multi-threaded communication library in said data processing system;

a thread dispatcher in said data processing system for efficiently dispatching threads awaiting messages in said multi-threaded communication library;

computer code which preassigns threads to messages to be received;

computer code which puts to sleep those threads whose assigned messages have not been received;

computer code which, upon receipt of a message, awakens its preassigned thread; and

computer code which executes said awakened thread, thereby processing the received message and testing to see if any threads are ready to run.

38. The apparatus of claim 37 wherein the selection of the thread to be dispatched is based on its priority as set when the thread is put to sleep.

39. The apparatus of claim 37 wherein said computer code which preassigns threads comprises:

computer code which creates a thread-specific structure for each thread, each thread-specific structure having a ready flag and a condition variable unique to its preassigned thread;

computer code which creates a handle for each message to be received; and

computer code which causes a thread invoke message passing logic for a particular handle, thereby associating the thread and the message.

40. The apparatus of claim 39 wherein said computer code which puts to sleep comprises:

computer code which enqueues for a received message, a preassigned thread-specific structure into a first queue;

computer code which writes into said handle associated with the message received, an identification of said thread-specific structure enqued for the received message, and

computer code which places said thread-specific structure for the received message in the WAIT condition.

41. The apparatus of claim 40 wherein said computer code which awakens comprises;

computer code which completes said received message;

computer code which changes the condition of the thread-specific structure for the completed received structure to the READY condition; and

computer code which dequeues with a queue manager, the next thread-specific structure in said first queue in the READY condition and sending its thread a thread awakening condition signal.

42. The apparatus of claim 41 further comprising;

in said computer code which preassigns, computer code which allocates buffer space for storing messages to be received; and

in said computer code which puts to sleep, computer code which identifies in said handle the buffer in which the message associated with the handle is to be stored when it is received.

43. The apparatus of claim 42 wherein said computer code which completes said received message comprises computer code which stores said received message in the buffer identified in the associated handle for the received message.

44. The apparatus of claim 41 wherein said queue manager includes computer code which dequeues the next thread-specific structure using a First-In-First-Out policy.

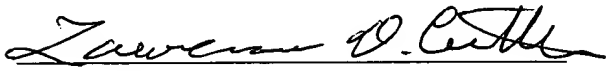
45. The apparatus of claim 41 wherein said queue manager includes computer code which dequeues the next thread-specific structure using a Last-In-First-Out policy.

46. The apparatus of claim 41 wherein said queue manager includes computer code which dequeues the next thread-specific structure based on a priority value contained in said structure.

47. The apparatus of claim 41 further comprising computer code which obtains a lock for the handle associated with said received message such that the awakened thread may process only the received message.

48. The apparatus of claim 47 further comprising computer code which releases said lock after said awakened thread has processed said received message such that said awakened thread may continue with other work.

RESPECTFULLY SUBMITTED

Date: March 28, 2005 By: 

Lawrence D. Cutter, Senior Attorney
Registration No. 28,501
Phone: (845) 433-1172
FAX: (845) 432-9786

OIPE JC
MAR 31 2005
PATENT & TRADEMARK OFFICE

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No. POU919990100US1

In Re Application Of: **Tuel et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/588,492	06/06/2000	M. Banankhah	33558	2127	9151

Invention: **THREAD DISPATCHER FOR MULTI-THREADED COMMUNICATION LIBRARY**

COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on **01/26/2004**

The fee for filing this Appeal Brief is: **$500.00**

- ☐ A check in the amount of the fee is enclosed.
- ☐ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **09-0463**
- ☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Lawrence D Cutter
Signature

Dated: **March 28, 2005**

LAWRENCE D. CUTTER, Senior Attorney
Reg. No. 28,501
IBM Corporation
2455 South Road, M/S P386
Poughkeepsie, NY 12601
(845) 433-1172

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

03/28/05
(Date)

Susan D Phelps
Signature of Person Mailing Correspondence

SUSAN L. PHELPS
Typed or Printed Name of Person Mailing Correspondence

cc: